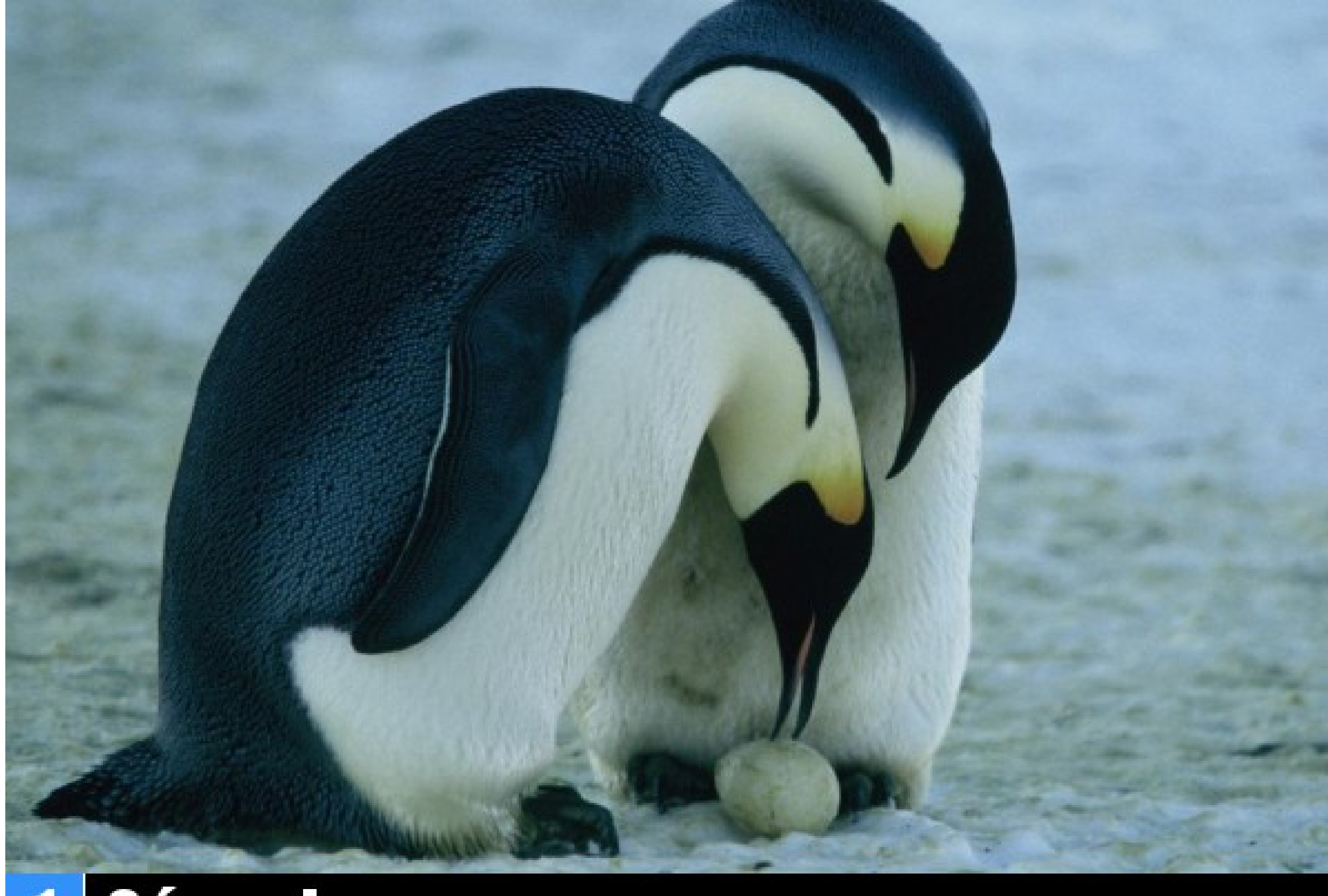


1 Génesis

Revista de Software Libre ATIX 2008

Colaboradores

Equipo Editorial

Dirección y Coordinación General

Esteban Saavedra López (jesaavedra@opentelematics.org)

Diseño y Maquetación

Jenny Saavedra López (jennysaavedra@hotmail.com)

Esteban Saavedra López (jesaavedra@opentelematics.org)

Colaboración

Williams Chorolque Choque (williamsis@gmail.com)

Herramientas

La edición de esta revista fue realizada de forma integra haciendo uso de Software Libre







El que lo sabe, el que lo intenta, el que lo puedo, el que lo logra

Editorial

Génesis

....Y así comenzó la historia de la revista **ATIX**.

Se dice que un hombre debería hacer 3 cosas importantes en su vida: tener un hijo, plantar un árbol y escribir un libro; con un poco de analogía podríamos comenzar a contar la historia de la comunidad **ATIX**, que empezó no teniendo un hijo sino varios, hijos que durante este tiempo le brindaron su tiempo y colaboración, poca o mucha no interesa, lo importante es que colaboraron.

ATIX como buen padre conjuntamente a sus hijos fueron poniendo las semillas y plantines, que en un futuro cercano se convertirán en los árboles, árboles cuyas ramas se expandirán con efecto viral, árboles que crecerán en busca de lograr su objetivo, árboles que perdurarán en el tiempo, árboles que contarán su historia.

No a través de un libro pero si de una revista, empezamos a mostrar lo mucho o lo poco que aprendimos, lo mucho o lo poco que sabemos, lo importante es lo que nosotros podemos y queremos mostrar, puede ser que para muchos sean trabajos sencillos y triviales, para nosotros son trabajos con mucho valor, por que denotan el esfuerzo y sobre todos las ganas de querer colaborar.

ATIX de esta forma, no esperó a crecer mucho, en muy poco tiempo, tiene varios hijos, planto muchos árboles y ha escrito su revista.

Como líder de esta comunidad, me siento muy complacido de presentar el primer número de la revista **ATIX**, en cuya estructura hemos tratado de contemplar varios aspectos importantes, algunos que se fueron perdiendo dentro nuestra sociedad:

- ✔ El de brindar un espacio a todos los que deseen aportar, no solo aquellos con habilidades extraordinarias, sino realmente a todo aquel que tenga ganas de colaborar.
- ✔ El brindar un apoyo a la gente que recién comienza, por que consideramos que todos tuvimos un empiezo y todos aprendimos desde lo más básico.
- ✔ Tratar de motivar e incentivar a que no solo seamos consumidores de conocimiento y tecnología existente, más al contrario tratemos de realizar aportes al conocimiento y tecnología existente.
- ✔ En simples palabras “**Fomentar lo nuestro**”

Por último, esperamos que nuestro trabajo sea del agrado de todos nuestros lectores, comprometidos estamos a hacer que este proyecto sea un aporte continuo a toda la comunidad de Software Libre dentro y fuera de nuestras fronteras.

Esteban Saavedra López
Director y Coordinador General

Génesis

Contenido



Palabra quechua,
con un sentimiento profundo
y con un gran significado filosófico

El que lo sabe

El que lo intenta

El que lo puede

El que lo logra

Programas equivalentes Winbugs-Linux

Generalmente una persona que está comenzando con el Software Libre no tiene idea de cuales pueden ser los equivalentes de los programas que utiliza habitualmente en Winbugs, para que decida de una vez cambiarse por completo a Linux. Este artículo pretende mostrar algunas de las muchas alternativas que disponemos en la actualidad.



Lo mas usual que utilizamos en nuestro trabajo diario son: trabajos en ofimática (Word, Excel,PowerPoint), escuchar música (winap, bspalayer), navegar por internet (Internet Explorer), Quemar Cd/Dvd (Nero), reproducir películas o vídeos (Winbugs media player), visualizar documentos PDF e imágenes (Acrobat Reader,), chat (Mirc), conversar en línea(Msn), etc.
Estas son las tareas que un usuario normal realiza con estos programas diariamente y por ende necesitamos algo igual para empezar a trabajar. Ahora veremos que no son solo similares sino mucho mejores en funcionamiento, configuración y rapidez.

Ofimática



Para el manejo y creación de documentos la mejor alternativa que recomiendo es Openoffice que incluye herramientas importantes como:

- **Writer:** El procesador de texto equivalente a MS Word.
- **Calc:** Permite generar y manejar hojas de calculo equivalente a MS Excel.
- **Draw:** Presentaciones, equivalente a MS Power Point.
- **Base:** Equivalente A MS Access, para base de datos.

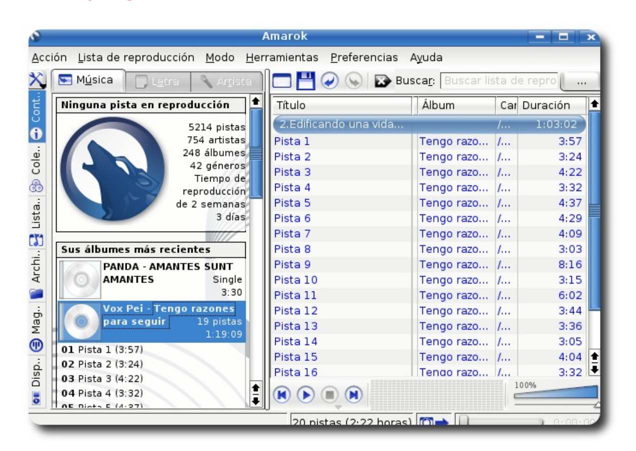
Reproductores de audio y video

XMMS



Si alguien me preguntara cual es el mejor reproductor audio de Linux respondería sin dudar
que este es este XMMS, es muy liviano y parecido al Winap. Además cuenta con muchísimas pieles para los mas exigentes

Amarok



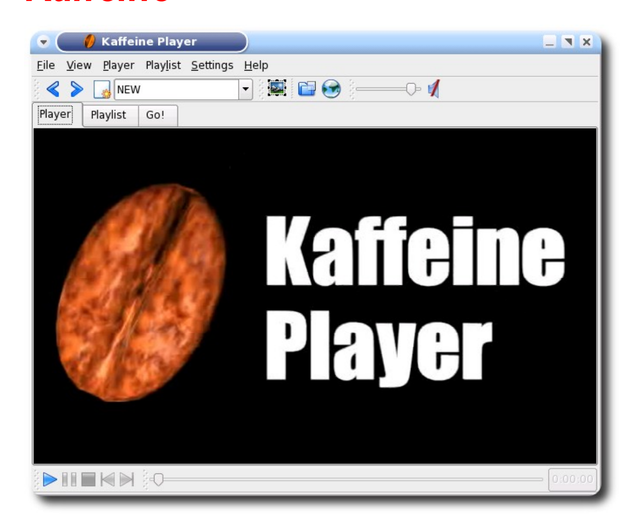
Como olvidar a nuestro querido amarok poderoso reproductor de Linux por excelencia, realmente es de lo mejor, necesita un poco más de recursos de nuestra PC, también se puede obtener pieles para ésta aplicación, tan solo tienen que probarla y verán la enorme diferencia con las aplicaciones para Winbugs.

Mplayer



Reproductor multimedia, reproduce archivos MPEG, VOB, AVI, OGG/OGM, VIVO, ASF/WMA/WMV, QT/MOV/MP4, FLIP, RM, Film, RoQ, soportados por algunos códecs nativos Xanim y DLL's, Win32, además puede reproducir VideoCd, SVCD, DVD, 3ivx y Divx 3/4/5.

Kaffeine



Este es mi favorito ha sido un compañero inseparable a la hora de ver mis películas en DVD, y sobre todo un sin fin de animes, se los recomiendo usarlo es bastante liviano y con muchas opciones de configuración y reproduce bastantes formatos de vídeo.

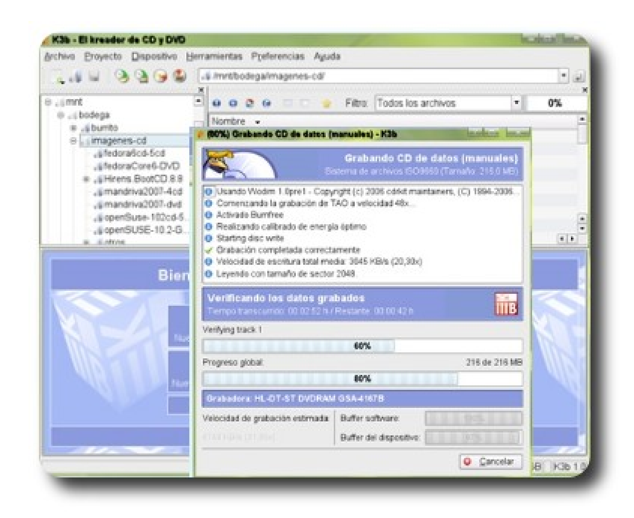
VLC



Si alguno de los dos reproductores mencionados arriba les llega a fallar con algún formato raro ¡Dudo mucho!, les aseguro que este reproductor no los va a decepcionar es excelente y también liviano.
Que esperas para cambiarte a LINUX!!!!!!, y son solo los más importantes, existen reproductores de audio y vídeo para todos los gustos y exigencias y lo mejor, totalmente libre, no tienes que pagar ni un solo boliviano por obtenerlo.

Quemando Cd /DVD en Linux

K3B



Cual Nero, aquí con toda autoridad el que manda en Linux es k3b excelente interfaz gráfica para la grabación de discos compactos y DVD, permite entre otras cosas la grabación de Cd-audio, creación de vídeo Cd, copia exacta de cd (copia clon), grabación de DVD de datos, creación de vídeo DVD. Extracción de Cd/DVD en imagen ISO, permite grabar también esta imagen ISO en CD/ DVD. Por si fuera poco también cuenta con pieles para su configuración.Brasero



Una alternativa más para la grabación de CD/DVD. es más liviano no ocupa muchos recursos, una interfaz sencilla de manejar, eficaz y rápida.

Internet, chat, conversación en linea

Mozilla Firefox



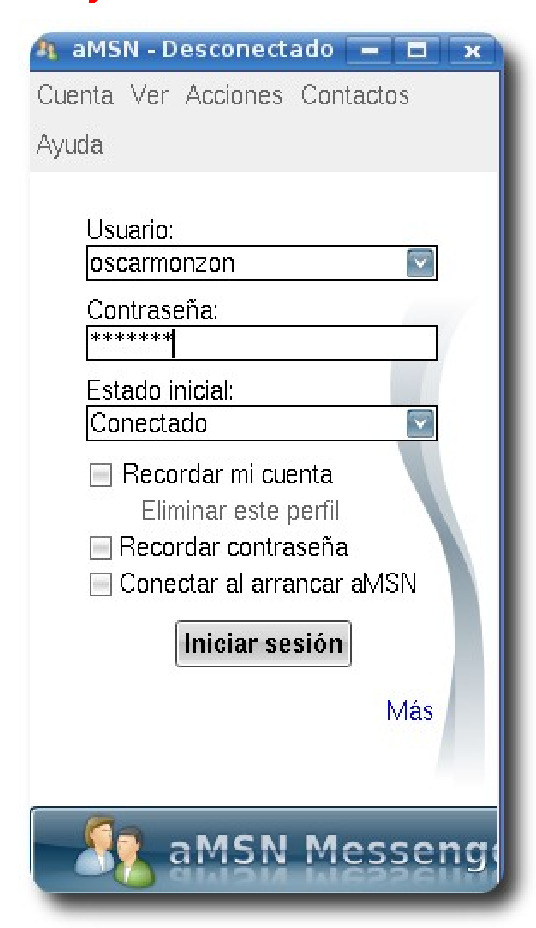
Excelente y el mejor navegador no lo digo yo, revisen en internet cual es el más usado sin dudas de lejos es este. navegación por pestañas, plugins hará casi todo lo que uno necesite.

Konqueror



Excelente navegador, también funciona como visor y gestor de archivos, es muy interesante este proyecto es como si uno estuviera navegando en la internet, sin embargo es su ordenador.

aMsn y Xchat

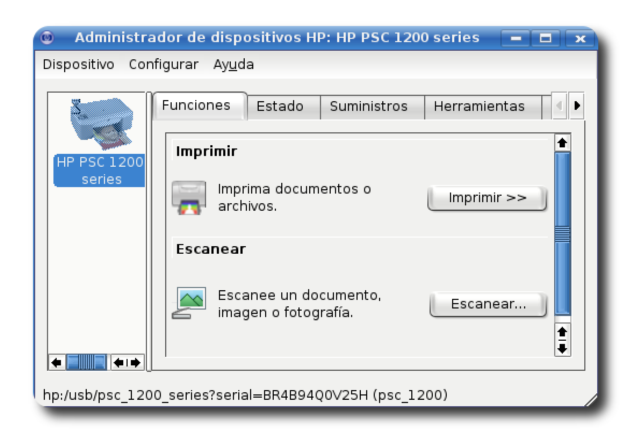




No hay mucho de que hablar de estos 2 programas su funcionamiento como ya se imaginarán, es simliar al Msn y al Mirc (que funcionan en Winbugs)

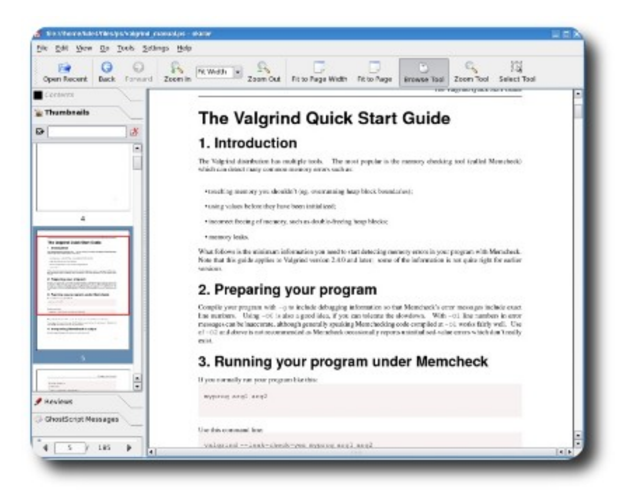
Impresora y documentos en pdf

HPLIB



Es un tema un poco conflictivo hacer reconocer la impresora en Linux, imaginense el escáner, pues a no preocuparse para los que utilizan estas dos en la marca HP (Hawlet Packard), este programa te facilita la utilización de ambas, tienes la posibilidad de configurarla exactamente como si estuvieras en Winbugs, en: Calidad, que cartucho utilizará, tipo de papel,etc.

Evince Kpdf



Estos programas son muy similares al conocido Acrobat Reader, así que no tendrán ningún problema al hacer el uso y manejo de ambos programas y vuelvo a recalcar totalmente libres.

Referencias

Ha sido un grato placer haberles traído este pequeño documento, que ojo fué hecho en OO, acá les dejo dos direcciones donde encontramos más equivalencias:

http://www.linuxalt.com
http://alts.homelinu.net

Autor

Oscar Choque Monzón
Estudiante de Ingeniería Informática
oscar.ch.monzon@gmail.com

Conociendo Vim
Guía de Supervivencia

Este es un artículo introductorio al uso de Vim un editor de texto en modo consola muy potente y extensible además que se encuentra en cualquier versión de GNU/Linux.

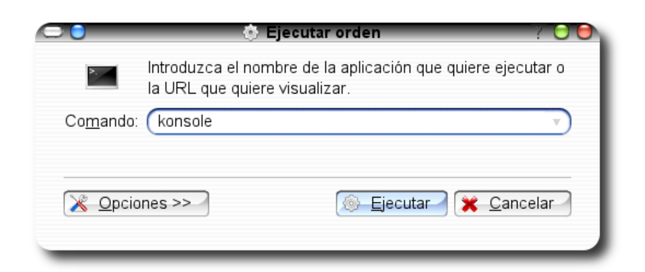


Hoy en día parece algo extraño y arcaico el uso de la consola y aplicaciones orientadas a la consola siendo que nos hemos mal acostumbrado a depender del mouse siendo que cuando nos encontramos en un ordenador sin mouse no sabemos que hacer ni por donde empezar, pero una particularidad (muy buena por cierto) de GNU/Linux es el uso intensivo de la consola donde podemos encontrar desde editar documentos de texto plano, escuchar música, ver vídeo, Documentación de Proyectos de manera Profesional (Uso de LaTeX en alguna ocasión realizaremos un artículo sobre ello), en fin son muy pocas las cosas que no podemos hacer en la consola, justamente comenzando a ver como podemos trabajar con ella y dejando descansar el mouse es que vamos a conocer un poco de como movernos con vim uno de los mejores editores que poco o nada tiene que envidiar a los editores modernos (lo de modernos en términos de actualidad) existentes hoy en día.

En este artículo vamos a trabajar en la consola pero como podemos abrirla? Bueno dependiendo de si estamos en KDE o GNOME la cosa varia, pero algo común es la combinación de teclas **Alt+F2** ahora introducimos el siguiente comando

para usuarios de KDE
konsole

para usuario de GNOME
gnome-terminal



con esto ya tendremos disponible una consola listo para trabajar.

¿Que es VIM?

Vim (del inglés Vi IMproved) es una versión mejorada del editor de texto vi, presente en todos los sistemas UNIX. Su autor,Bram Moolenaar, presentó la primera versión en 1991, fecha desde la que ha experimentado muchas mejoras. La principal característica tanto de Vim como de Vi consiste en que disponen de diferentes modos entre los que se alterna para realizar ciertas operaciones, lo que les diferencia de la mayoría de editores comunes, que tienen un sólo modo en el que se introducen los comandos mediante combinaciones de teclas o interfaces gráficas.

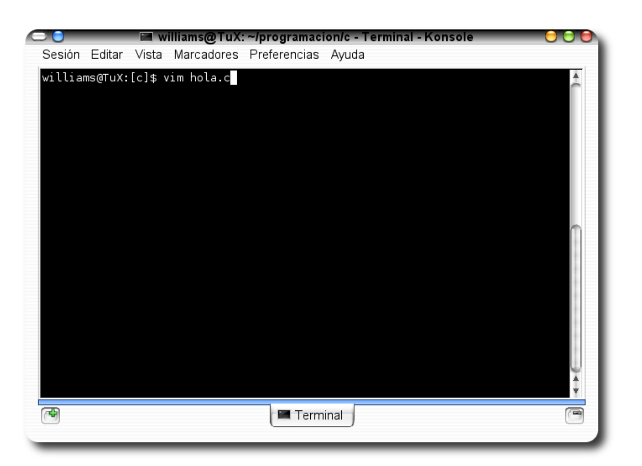
Instalando VIM

Si bien vim se encuentra en cualquier distribución de GNU/Linux es necesario que este correctamente instalado por lo tanto no nos vendría mal ejecutar el siguiente comando en la consola como usuario

administrador (root) **apt-get install vim** esto para usuarios de la Distribución Debian o basado en Debian como ubuntu, kubuntu, knoppix, Tuquito, para los usuarios que trabajen en una distribución basado en **Red Hat** como fedora tendrán que realizarlo usando el siguiente comando **yum install vim**.
En caso de que la instalación ya este correcta y completa no se necesitara actualizar/instalar ningún paquete.

Iniciando VIM

Una vez en la consola podemos iniciar de varias maneras que a continuación detallamos el símbolo de "$" es el que aparece en la consola así que no la introduzcan.



$**vim**	Editar un texto sin nombre
$**vim** archivo	Editar un archivo (nuevo o no)
$**vim** archivo1 archivo2	Editar lista de archivos
$**vim** +n archivo	Editar archivo y posicionarse automáticamente en la línea n.
$**vim** +/txt archivo	Editar archivo y posicionarse en la 1a línea donde aparece txt

Modos de uso de VIM

Cuando trabajemos en vim podemos encontrarnos en alguno de los siguientes modos

- **Modo Edición o Inserción**
  En este modo podemos llevar a cabo la edición del documento pero no es el por defecto así que tenemos que hacer algunas cosillas para entrar en este modo.
- **Modo de Comando**
  Este es el modo por defecto en que arranca vim y como su nombre indica necesitamos introducir algunos comandos por ejemplo tecleando **i** ingresaremos el modo edición/inserción si en este modo tecleamos ESC (la tecla ESCAPE) volveremos al modo comando.
- **Modo línea de comando**
  Este modo es similar al anterior solo que podemos hacer uso de este con comandos un poco más complejos y es necesario colocar **:** (dos puntos) antes de ejecutar cada comando.

Si bien parece muy complicado en este momento y más de uno se este desanimando a usar vim no lo haga que muy pronto verá que usar vim no es tan difícil como parece solo es cuestión de acostumbrarse un poco y disfrutar la potencia de este editor.

Comandos básicos

Como les dije anteriormente el modo por defecto de vim es el modo comando así que veamos algunos comandos interesantes de vim

i Inicia inserción en la posición del cursor.

I Inicia inserción al principio de la línea.

a Inicia inserción después de la posición actual del cursor.

A Inicia inserción al final del línea

o Inserta una línea después de la línea actual y comienza el modo de inserción.

O Inserta una línea antes de la línea actual y comienza el modo de inserción.

Si nos encontramos en modo inserción/edición podemos volver fácilmente al modo comando pulsando la tecla de **ESCAPE** (**Esc** de ahora en adelante).

Editando documentos

Al editar un documento rara es la ocasión en la cual no se necesite copia pegar o borrar algo y la pregunta que surge es ¿como hago esto?. Podemos realizarlo de la manera tradicional usando BackSpace y terminando odiando el editor o haciendo uso de los comandos de vim. ¿Más comandos todavía? Si todavía más comandos pero muy útiles ya que agilizaran tu trabajo estos los debes realizar en modo comando; presiona **Esc** si te encuentras en modo edición/inserción.

0 La letra 0 (cero) nos permite movernos automáticamente al inicio de la línea actual.

$ Con este comando nos movemos automáticamente al final de la linea, te diste cuenta de como ahorraste tu tiempo al usar estos 2 comandos en vez de usar la flecha izquierda/derecha?

dd A borrar se dijo con este comando podrás hacer desaparecer una línea de un plumazo.

D No siempre necesitas hacer desaparecer toda la línea solo desde cierto lugar por lo tanto con este comando borraras desde la posición actual del cursor hasta el final de la línea.

w No siempre vamos de principio a final o viceversa a veces es necesario avanzar al inicio de la siguiente palabra para ese caso usamos este comando.

b Si podemos ir adelante también podemos ir atrás palabra por palabra con este comando.

J Con este comando podemos unir al linea actual con la siguiente.

x Algún carácter indeseable? Pues coloca con el cursor sobre dicho carácter y borrarlo con este comando

/cadena o **?cadena** Dicen que el que busca encuentra nada más cierto con este comando, haciendo uso de **/cadena** (siento “cadena” la palabra que buscas) podrás encontrar (si es que existe) la siguiente aparición de dicha cadena desde la posición actual del cursor, con **?cadena** lo encontrarás desde la posición actual hacia atrás.

n Que pasaba si deseas volver a realizar la búsqueda nuevamente? Pues no es necesario volver a introducir el comando /cadena solo presiona la tecla **n** y todo listo y así cuantas veces necesites.

N encontramos la ocurrencia anterior del texto buscado

yy Si bien con dd eliminábamos la línea actual con yy copiamos toda la línea.

p Con este comando pegamos lo que hayamos copiado con yy o borrado con dd en general lo último que hayamos copiado o borrado. La copia se la hace debajo de la línea actual.

P Es similar a **p** pero la copia se lo hace en la línea anterior a la línea actual.

u A deshacer se dijo y es que con este comando podemos deshacer las últimas modificaciones; este es el clásico undo

. Si podemos deshacer con u también podemos rehacer y esto lo realizamos con **.** (punto).

Modo línea de comandos

Si bien hasta el momento ya estamos listo para poder editar un archivo de texto cualquiera, hasta el momento no vimos como guardar, abrir, salir, etc en vim, para ello necesitamos estar en modo línea de comandos por lo tanto lo único que tenemos que hacer es presionar **Esc** si estamos en modo edición/inserción caso contrario no es necesario hacer nada.
Veamos que comandos tenemos en este modo:

```
Tetris.py (~/programacion/python) - VIM - Terminal - Konsole
Sesión  Editar  Vista  Marcadores  Preferencias  Ayuda
 1 #!/usr/bin/python
 2
 3 # Filename : Tetris.py
 4
 5 import sys
 6 import random
 7 from PyQt4 import QtGui
 8 from PyQt4 import QtCore
 9
10 class Tetris(QtGui.QMainWindow):
11     def __init__(self):
12         QtGui.QMainWindow.__init__(self)
                                 10,1      Comienzo
Terminal
```

```
hola.c (~/programacion/c) - VIM - Terminal - Konsole
Sesión  Editar  Vista  Marcadores  Preferencias  Ayuda
  1 #include <stdio.h>
  2 int main()
  3 {
  4   printf("Hola Mundo\n");
  5   return 0;
  6 }
~
~
:e ~/programacion/python/Tetris.py
Terminal
```

```
help.txt - (ayuda) - VIM - Terminal - Konsole
Sesión  Editar  Vista  Marcadores  Preferencias  Ayuda
help.txt       For Vim version 7.1.  Last change: 2006 Nov 07

                     VIM - main help file
                                                                k
     Move around:  Use the cursor keys, or "h" to go left,     h   l
                   "j" to go down, "k" to go up, "l" to go right.  j
Close this window:  Use ":q<Enter>".
   Get out of Vim:  Use ":qa!<Enter>" (careful, all changes are lost!).

help.txt [Help][RO]                               1,1        Comienzo
  4
  5 import sys
  6 import random
  7 from PyQt4 import QtGui
  8 from PyQt4 import QtCore
  9
 10 class Tetris(QtGui.QMainWindow):
 11     def __init__(self):
~/programacion/python/Tetris.py                   9,0-1          0%
Terminal
```

:w Bueno después de editar nuestro documento lo que necesitamos es guardarlo para lo cual usamos este comando, si al momento de llamar a vim lo hicimos si el parámetro del nombre del archivo aquí le podemos dar el nombre por ejemplo :w documento.txt caso contrario solo con :w es suficiente.

:e archivo No siempre trabajamos con un solo documento siempre nos encontraremos editando más de un archivo por lo tanto para poder abrir un archivo sin necesidad de salir de vim este es el comando que debemos usar.

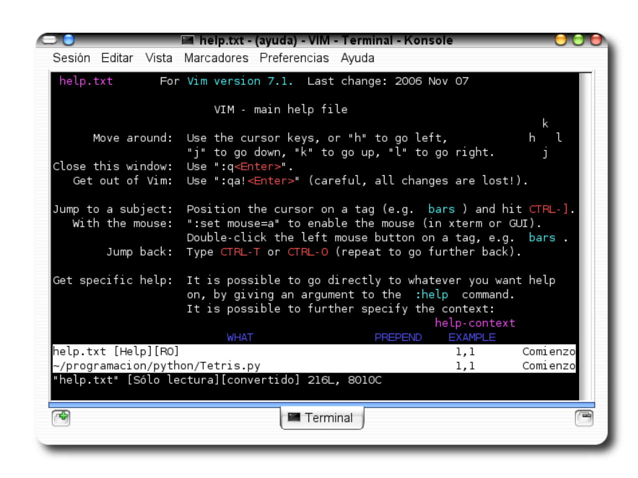
:q Después de una intensa jornada de trabajo con **:q** podemos salir de vim pero ojo es necesario guardar antes el archivo, ahora si queremos salir sin guardar

solo colocar **:q!** y todo listo.

:wq Como ya se lo habrán imaginado aquí guardamos y salimos un equivalente de esto es **:x** o **ZZ**.

:s/txt/txt2 Con esto buscamos la primera cadena que sea igual a txt y la cambiamos a txt2 si añadimos **/g** al final reemplazamos todas las palabras que sean igual a txt por txt2 otra variante es **:m,n s/txt/txt2 /g** en la cual la sustitución se la hace desde la línea **m** hasta la línea **n**.

:help Bueno creo que el nombre lo dice todo nada como una buena ayuda.



Bueno por el momento tenemos muchos comandos para prácticar y para empezar a usar y disfrutar de vim si bien al comienzo puede parecer aburrido y cansador tantos comando veremos que con un poco de práctica se convertirá en uno de nuestros editores favoritos.

En el siguiente número veremos como abrir más de un archivo a la vez, dividir nuestra ventana de manera vertical y horizontal, Expandir y colapsar bloques de texto (muy útil a la hora de programar), hacer uso de tabs y muchas otras cosas más, así que hasta el siguiente número.

Sitios de Interés

http://www.vim.org
http://es.wikipedia.org/wiki/Vim

Autor

Williams Israel Chorolque Choque
Estudiante de Ingeniería Informática
email: williamsis@gmail.com

Tuquito 2.0
Dale una oportunidad

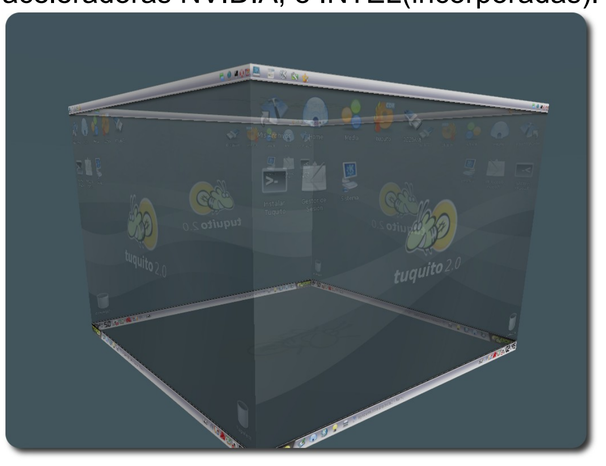
Este es un artículo que muestra una pequeña descripción de la distribución Tuquito, mostrando algunas de sus características..

Por que darle una oportunidad?



Tuquito es una distribución GNU/Linux, que implementa la tecnología LiveCd (funciona desde el Cd), esto esta muy de moda en el ambiente de las diferentes distros que Usted conoce, pero TUQUITO nos da la ventaja que tenemos en cuestión de minutos un sistema completo, con todo el Software y Hardware configurados y listos para usar, sin necesidad de modificar en absoluto su computadora, ganando tiempo a la hora de realizar tareas concretas y para nuestro medio que pocos son las personas que tienen un acceso a internet ADSL

Para los aficionados en los escritorios en 3D esta versión viene con Compiz, y por si fuera poco nos trae controladores para tarjetas aceleradoras NVIDIA, e INTEL(incorporadas).



El uso de Tuquito no requiere de ningún nivel de experiencia, esta destinado a todos tipo de usuarios. También puede instalarse en su computadora mediante dos tipos de instalación, Completa o Básica, teniendo todo configurado y listo en su disco rígido en un tiempo mínimo.



Tuquito cuenta con 2 Gigabytes de

aplicaciones en un cd común de 650Megabytes, con una amplia selección de paquetes en las áreas de ofimática, ingenierías, multimedia, internet, programación y ciencias.
La estética instantánea de Tuquito, permite disfrutar de un sencillo y amigable entorno de trabajo.



Algunas características

- Detección automático de los discos rígidos de su computadora, dejándolos disponibles en su escritorio con tan solo un click.
- Soporte de variadas impresoras de modelos, antiguos y nuevos, disponibles actualmente en el mercado.
- Soporte de scanners.
- Soporte de webcams, incluyendo los modelos de intel, genius.
- Permite acceder a cdroms, diskettes, con tan solo un click en su escritorio.
- Detección automática de cámaras digitales, pendrives, dispositivos hotplugs(usb).
- Es compatible con archivos de MSOFFICE y soporta los sistemás de archivos de Windows (Fat32 - NTFS).

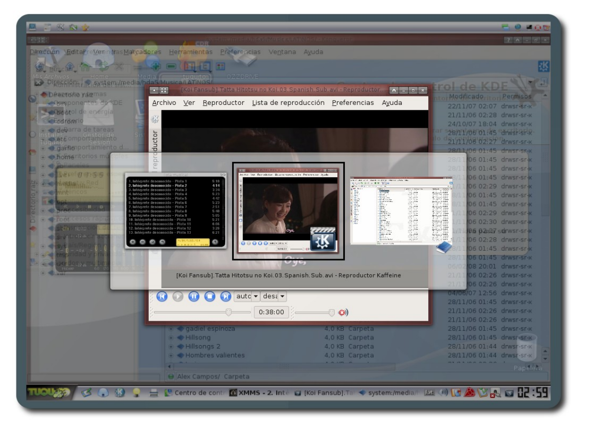
Ahora vamos a conocer a Tuquito un poco más, una frase dice que **una imagen vale más que mil palabras**.

Efectos 3D









Ventajas

- Montón de programás disponibles en tan solo 1Cd.
- Escritorio en 3D con el manejador de ventanas compiz
- Documentación del software en español.
- Reconoce una cantidad de Hardware bastante grande.
- Una interfaz gráfica retocada muy elegante.
- No nos exige muchas características de Hardware para su instalación.
- Controladores para tarjetas de vídeo NVIDIA, o sea tenemos disponible OpenGl, y aceleración 3D.
- Realmente rapidísimo a comparación de otras distros.
- El gobierno Argentino incluso lo tomo como SW base para las computadoras OLPC.

Desventajas

- Problemás de dependencias con algunos programás, por ejemplo Cedega(Para juegos corre muy bien en Ubuntu).
- Poco soporte por lo menos en Bolivia, pues es una distro Argentina.
- No encontramos mucha documentación por lo mencionado anteriormente.
- No es una de las distros más populares y por eso no cuenta con muchos foros y apoyo.
- Es relativamente nueva y pequeña comparada con la más popular de estos días Ubuntu (Se ha convertido en un monstruo del SW libre).

Conclusión

Tuquito es una distro muy buena para comenzar dentro de lo que es el Software libre, yo la recomiendo a las personas que aún están indecisas de que distro utilizar, por todas las ventajas ya mencionadas.
Respecto a las desventajas con una buena conexión a internet ADSL, se solucionarían una gran parte, pero seamos pacientes y esperemos que el servicio mejore en calidad y precio en nuestro medio.
Si alguno desea obtener esta distro, tan solo solicitenlo al grupo ATIX y encantados les copiaremos el LiveCd.
Este mi pequeño aporte para la revista de la comunidad ATIX, espero traer nuevas noticias tal ves no de TUQUITO, pero el tiempo ya dirá.
Me despido y cualquier pregunta o interrogante aquí les dejo la página oficial de la distribución y mi correo para cualquier duda.

Referencias

http://www.tuquito.com.ar

Autor

Oscar Choque Monzón
Estudiante de Ingeniería Informática
oscar.ch.monzon@gmail.com

JasperReport + Ireport Generación de Reportes en Java

Este es un artículo pretende introducir a la generación de reportes; que serán utilizadas desde aplicaciones Java.

Introducción

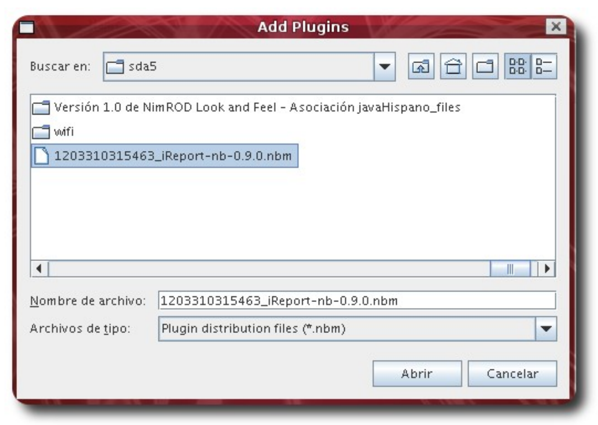
Un aspecto importante cuando se realiza el desarrollo de software, es la visualización de la información, ya sea con fines de toma de decisión, control de ítems (inventarios) u otros. Tal fuente de información por lo general casi siempre esta almacenada en Base de Datos o Archivos.

Hoy en día el software a desarrollarse esta obligado a desplegar reportes por pantalla como por ejemplo: la lista de trabajadores de una empresa, la cantidad de insumos de una tienda, etc. pero darle la funcionalidad de impresión complica un poco mas el desarrollo del software más aún usando como lenguaje de programación a Java, siendo esto antes una de las partes flacas de este lenguaje, cosa que en la actualidad a cambiado.

Buscando un poco me encontré con varias herramientas para cumplir este fín, una de éstas es: JasperReport, siendo Open Source y con la posibilidad de generar contenido por impresora y exportación a otros formatos como PDF, XSL, etc.

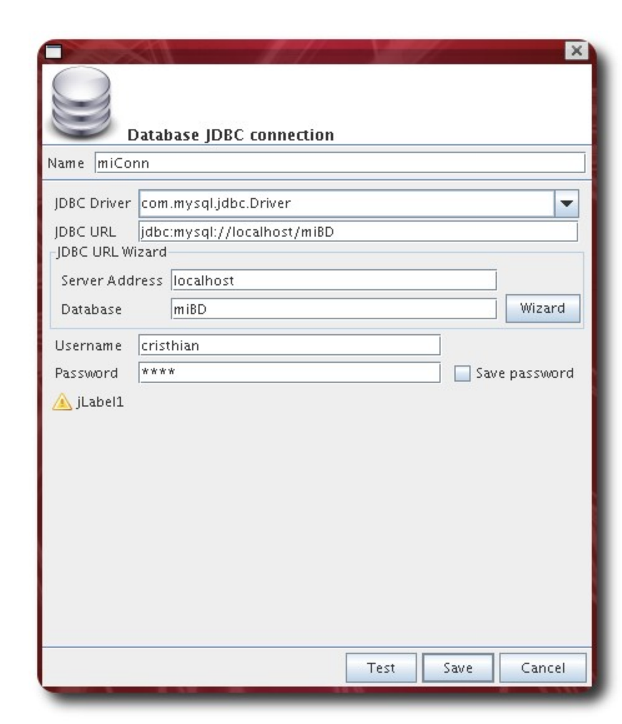
Por lo tanto en este articulo se mostrara como diseñar, compilar y ejecutar un reporte usando iReport , luego la forma de visualizarlo dinámicamente desde una aplicación , desde luego en Java
Para hacer un poco mas fáciles las cosas vamos a usar:

- ✔ `NetBeans 6.0`
- ✔ `Derby.jar`
- ✔
- ✔ `1203310315463_iReport-nb-0.9.0.nbm`



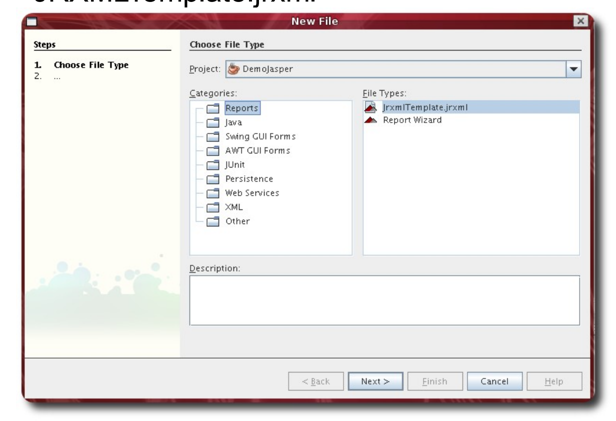
El primero es un IDE para el desarrollo en Java, el segundo es el drive del motor de Base de Datos Derby y el tercero es un plugin para NetBeans que contiene JasperReport + IReport, el cual se instála en NetBeans (Tool->Plugins->Downloaded->Add Plugins), se localiza el path de 1203310315463_iReport-nb-0.9.0.nbm y por último se clickea en Install, y ya tenemos a JasperReport + IReport como módulo en NetBeans.

Continuando vamos a proceder con la conexión a la Base de Datos: tras la instalación del plugin aparecerá el icono de Connections/Data Source, hacemos click en New para seleccionar una base de datos, damos todos los parámetros



Creación del Reporte

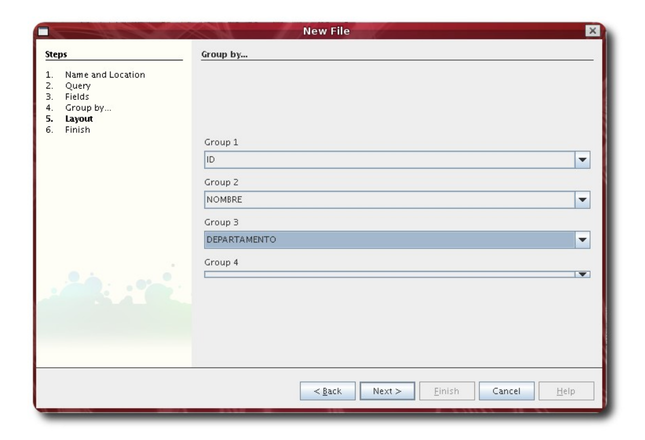
Vamos a crear un nuevo proyecto en File->New Project->Java Class, esto para empezar a diseñar el reporte, primeramente creamos en File->New File->JRXMLTemplate.jrxml



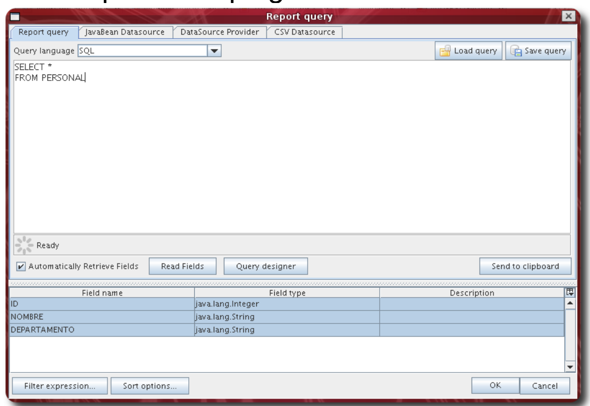
Posteriormente se visualizara el diseñador de reportes, identificando algunas secciones de este como: title, pageHeader, columnHeader, detail, etc

Diseño del Reporte

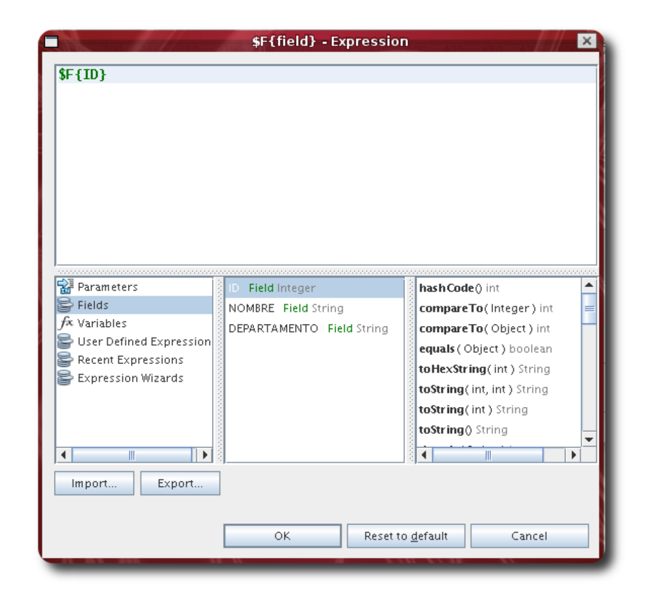
Ahora vamos a usar dos componentes para nuestro cometido, un Static Text y Text Field, arrastramos a la sección que deseemos, una vez hecho esto se procederá a modificar parámetros



Lo siguiente será realizar la consulta de de los campos a desplegar.

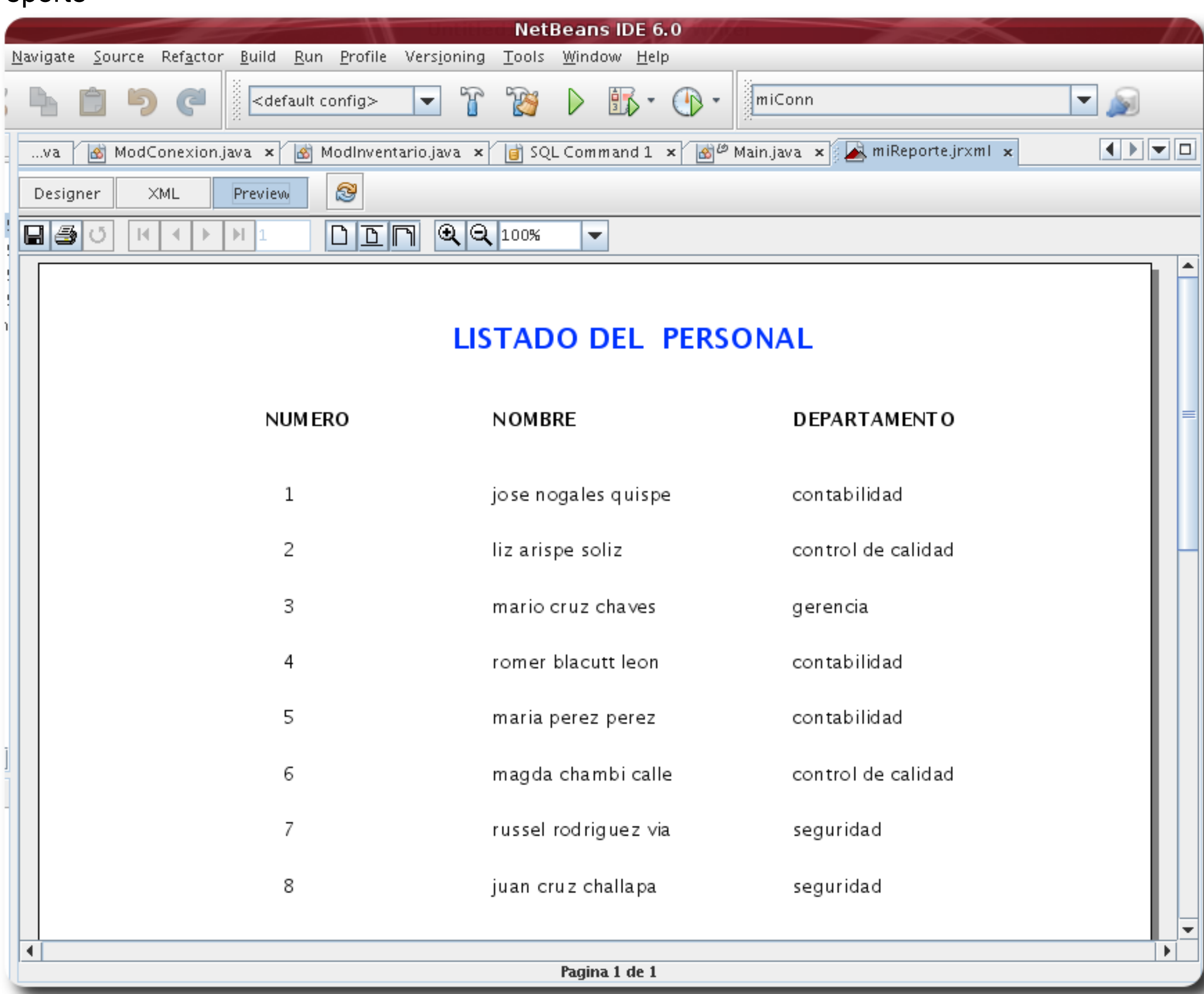


Se tendrá que realizar el cambio de parámetros de los TextField cambiando al nombre de campo que queramos visualizar como por ejemplo a $F{ID}. Por ultimo nos que hacer click en preview, desplegando un reporte de la siguiente manera:



Visualizarlo desde una Aplicación Java

Vamos a crear una nueva clase, en la cual vamos a escribir los métodos de visualización del reporte




```
//Este codigo  es el que ejecuta el reporte
package demojasper;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.HashMap;
import net.sf.jasperreports.engine.JRException;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.view.JasperViewer;
/**
 *
 * @author cristhian choquecallata
 */
public class VerReporte {
    Connection c;
    public VerReporte() {
        conexionDB();
    }
    void conexionDB(){
        try {
```

```
            Class.forName("org.apache.derby.jdbc.ClientDriver");
            c=DriverManager.getConnection("jdbc:derby://localhost:1527/miDB","cristhian
","cris");
        } catch (SQLException ex) {
           System.out.println("no se ha podido conectar con la BBDD");
        } catch (ClassNotFoundException ex) {
           System.out.println("no se ha podido realizar la carga del driver");
        }
    }
    void visualizar(){
        try {
            String ruta =
"/home/cristhian/NetBeansProjects/DemoJasper/src/demojasper/miReporte.jasper";
            JasperPrint jp = JasperFillManager.fillReport(ruta, new HashMap(), c);
            JasperViewer jv= new JasperViewer(jp,false);
            jv.show();

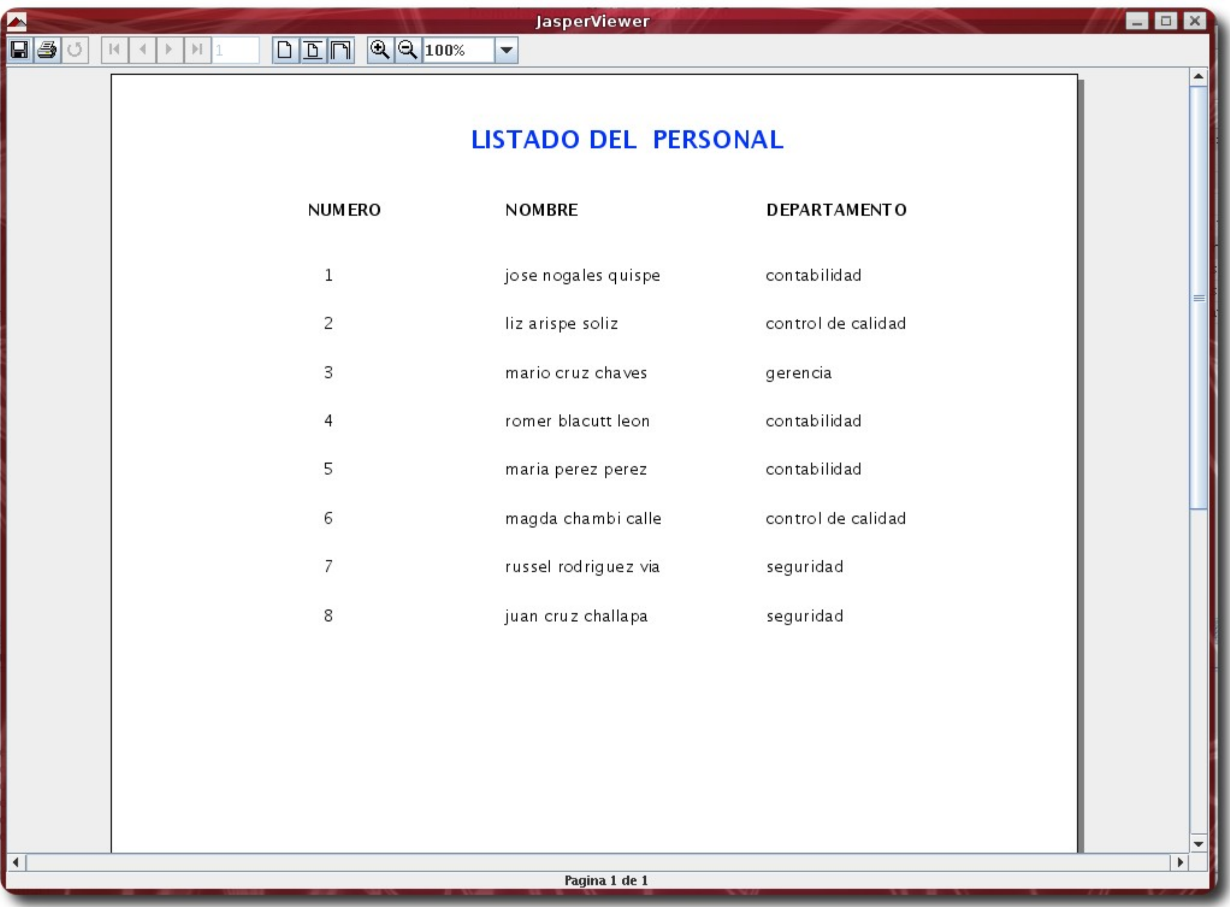
        } catch (JRException ex) {
        }

    }
}
```

Luego de terminar crear la clase desde una clase principal se tendrá que instanciar

```
      VerReporte x=new VerReporte();
             x.visualizar();
```

Compilamos el proyecto y ya esta tenemos una aplicación con el despliegue de reportes y la opción de exportar a formatos como PDF, XLS, y otros



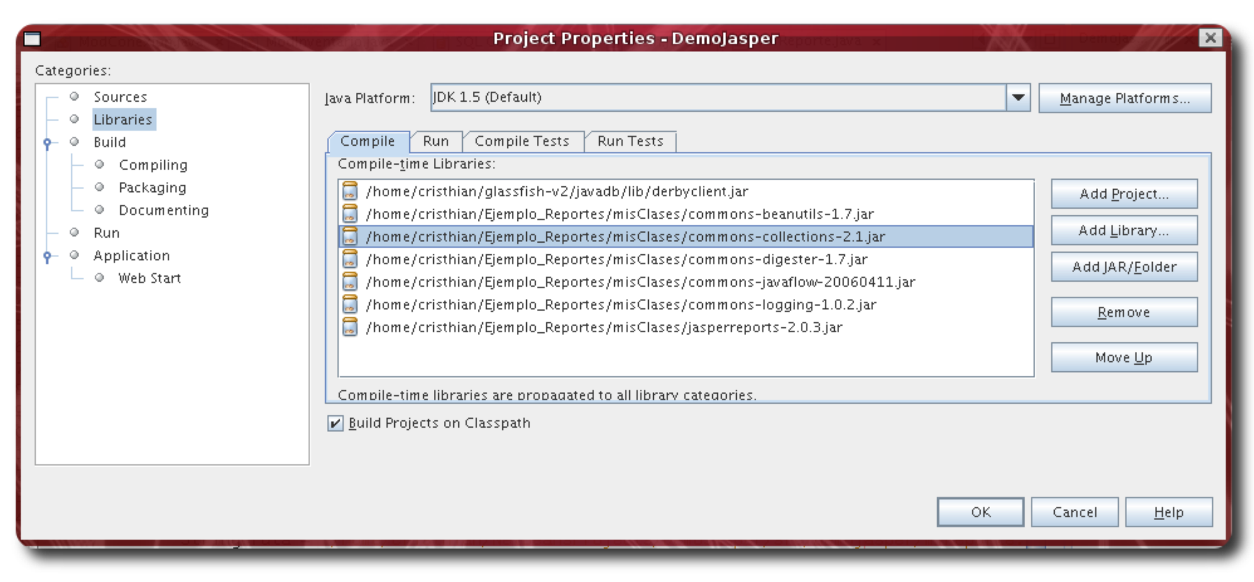
Para correr desde consola lo hacemos:

```
cristhian@cristhian-laptop: ~/NetBeansProjects/DemoJasper/dist
Archivo Editar Ver Terminal Solapas Ayuda
cristhian@cristhian-laptop:~$ cd NetBeansProjects/DemoJasper/dist/
cristhian@cristhian-laptop:~/NetBeansProjects/DemoJasper/dist$ java -jar DemoJas
per.jar
Vamos a desplegar el reporte
```

Conclusiones

Los reportes para su visualización necesariamente se necesita una versión del JDK, además de librerías que se añaden al proyecto como:

Y por último indicar que JasperReport no es la única opción a elegir a la hora de realizar reportes: encontrado a iText, Cristal Report, quedando a elección del programador.



Autor

Cristhian Choquecallata Machicado
Estudiante de Ingeniería Informática
email: cristhian_18_502@hotmail.com

Symfony Framework para el desarrollo de aplicaciones web

Symfony es un framework orientado al desarrollo de aplicaciones web de forma sencilla, rápida y divertida.

Introducción

Hoy en día, con la aparición de nuevas metodologías y herramientas en el ámbito de la ingeniería de software, el desarrollo de aplicaciones web ha tomado un cause un tanto distinto a lo que se tenía hace unos años atrás, ésto representa que, si antes con el desarrollo y herramientas tradicionales demorábamos semanas y/o meses en prototipiarlas, meses o hasta años en desarrollarlas completamente; hoy en día con la aparición de los frameworks éste tiempo se ha reducido considerablemente, representando que ahora con la ayuda de estas nuevas herramientas como son los frameworks y teniendo como aliado a las metodologías ágiles de desarrollo, podemos prototipear nuestras aplicaciones en cuestión de minutos u horas, y terminar su desarrollo en una par de semanas.

Que es un framework

- ✔ Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado.
- ✔ Se han convertido en la piedra angular de la moderna ingeniería del software.
- ✔ Es una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación.
- ✔ En otras palabras, es una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta.

Objetivos de un framework

- ✔ Desarrollo rápido.
- ✔ Desarrollo estructurado.
- ✔ Reutilización de código.
- ✔ Disminuir el esfuerzo en el desarrollo.
- ✔ Aprovechar las funcionalidades ya implementadas, no debemos reinventar la rueda.
- ✔ Nos concentramos directamente en la solución del problema.
- ✔ Tener como aliado a las metodologías de desarrollo ágiles (XP, AD).

Patrón MVC

Desde la aparición de RoR (Ruby on Rails) el padrino de todos los frameworks de desarrollo de aplicaciones web, el patrón MCV ha cobrado una notable importancia, sobre todo por su cualidad de dividir en capas el desarrollo de un proyecto o aplicación web.

- ✔ **Modelo:** Encargado de modelar y contener la lógica del dominio de la aplicación.
- ✔ **Vista:** Encargada de contener la lógica de presentación.

- **Controlador:** Encargado de ser el engranaje entre el modelo y la vista, y también contener la lógica del negocio.

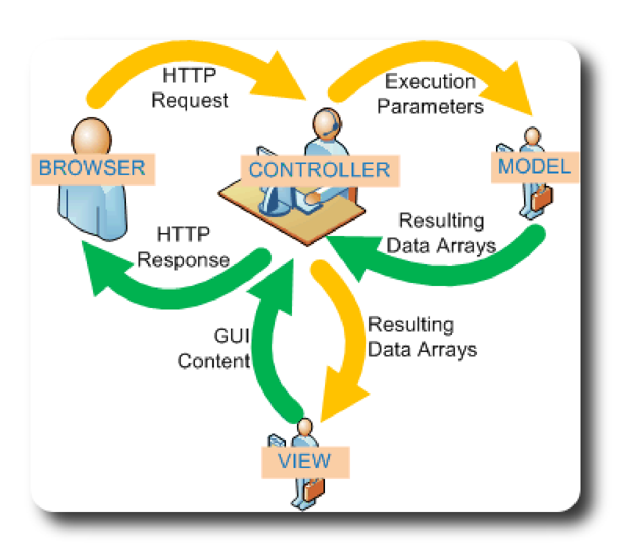


Gráfico 1: Patrón MVC

Que es Symfony

- Es un framework para el desarrollo de aplicaciones web.
- Es la respuesta de php para Ruby on Rails.
- Permite el desarrollo de aplicaciones web de forma rápida.

Filosofía de Symfony

- Reutilización de código.
- Desarrollar aplicaciones web de forma rápida, sencilla y divertida.
- Utilizar los mejores componentes actualmente disponibles.
- Combinar todo en un paquete fácil de usar.
- Symfony provee el pegamento, de tal forma que uds. no haga el suyo.

Características de Symfony

- Se basa en el patrón MVC.
- Tiene todo lo necesario para desarrollar aplicaciones Web.
- Soporta Ajax.
- Posee buena documentación.
- Es multiplataforma.
- Posee una numerosa comunidad de desarrolladores y usuarios.
- Inspirado en Ruby on Rails (Padrino de la mayoría de los frameworks).
- Está en constante desarrollo.

Componentes de Symfony

- **Propel:** ORM para el acceso a Base de datos
- **Creole:** Maneja la capa de abstracción de BD
- **Phing:** Mapeador XML
- **Pake:** Gestión de ejecución de Scripts

Preparando nuestro entorno de trabajo.

Para empezar a trabajar y desarrollar aplicaciones web en Symfony, precisamos cumplir algunos requisitos como ser:

- Tener instalado y configurado Apache o cualquier servidor de aplicaciones con soporte PHP.
- Tener instalado y configurado PHP 5o superior.
- Tener instalado algún gestor de base de datos (Sqlite, MySQl, PostgreSQL, MS-SQL, entre otros).

Instalación de Symfony

Utilizando Sandbox

- Obtener: http://www.symfony-project.com/get/sf_sandbox.tgz
- Basta con descomprimir y publicarlo en el servidor web

Mediante PEAR

```
✔ pear channel-discover
  pear.symfony-project.com
✔ pear remote-list -c symfony
✔ pear install symfony/symfony
```

Para confirmar que nuestra instalación es correcta, introducimos el comando Symfony -V, en la línea de comando.

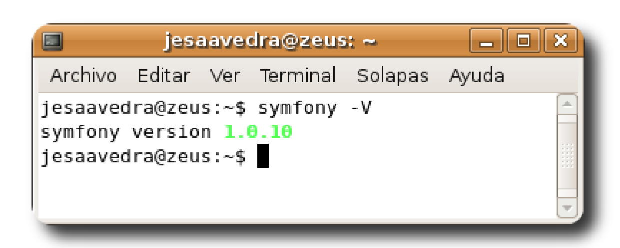


Gráfico 2: Uso del comando Symfony

Nuestra primera aplicación

Para una mejor comprensión del potencial de Symfony en el desarrollo de aplicaciones web, realizaremos una aplicación paso a paso.

Nuestra pequeña aplicación se enmarca en un entorno donde se mantiene el registro de eventos (congresos jornadas, etc), a los cuales concurren varios expositores de diferentes nacionalidades y donde cada uno de estos tiene una o más conferencias que impartir.

Estructura de un Proyecto

Hoy en día, los frameworks mantienen una forma similar de estructurar sus proyectos, imaginemos como si se tratase de un organigrama donde se consideran las siguientes partes:

- ✔ Un proyecto contiene una o varias aplicaciones
- ✔ Una aplicación tiene uno o varios módulos
- ✔ Un módulo tiene una o varias acciones
- ✔ Una acción puede o no estar asociada a una vista

Creación de un proyecto

La creación de un proyecto en symfony se la realiza mediante la instrucción:

```
symfony init-project aplicacion
```

Esta instrucción creará una estructura de directorios y dentro de ellos archivos que servirán para configuración, implementación y depuración del proyecto

Creación de una aplicación

La creación de un proyecto en symfony se la realiza mediante la instrucción:

```
symfony init-project aplicacion
```

Esta instrucción creará una estructura de directorios y dentro de ellos archivos que servirán para configuración, implementación y depuración de la aplicación

Estructura de directorios

Al igual que la mayoría de los frameworks orientados al desarrollo de aplicaciones web, Symfony cuando crea los proyectos y aplicaciones, por defecto crea una estructura de directorios definida, estructura que nos permite mantener un orden y conocer exactamente donde se encuentran todos y cada uno de los componentes de nuestra aplicación.

```
jesaavedra@zeus: ~/aplic_web/symfony/aplicacio
Archivo Editar Ver Terminal Solapas Ayuda
jesaavedra@zeus:~/aplic_web/symfony/aplicacion$ tree -d
.
|-- apps
|   `-- aplicacion
|       |-- config
|       |-- i18n
|       |-- lib
|       |-- modules
|       `-- templates
|-- batch
|-- cache
|-- config
|-- data
|   |-- model
|   `-- sql
|-- doc
|-- lib
|   `-- model
|-- log
|-- plugins
|-- test
|   |-- bootstrap
|   |-- functional
|   |   `-- aplicacion
|   `-- unit
`-- web
    |-- css
    |-- images
    |-- js
    `-- uploads
        `-- assets

29 directories
jesaavedra@zeus:~/aplic_web/symfony/aplicacion$
```

Gráfico 3: Estructura de la aplicación

Ejecutando nuestra aplicación

Hasta este momento nuestra aplicación no realiza ninguna tarea especifica, pero podemos observarla ejecutándose en algunos de sus entornos, por medio del browser, como muestra la siguiente figura:

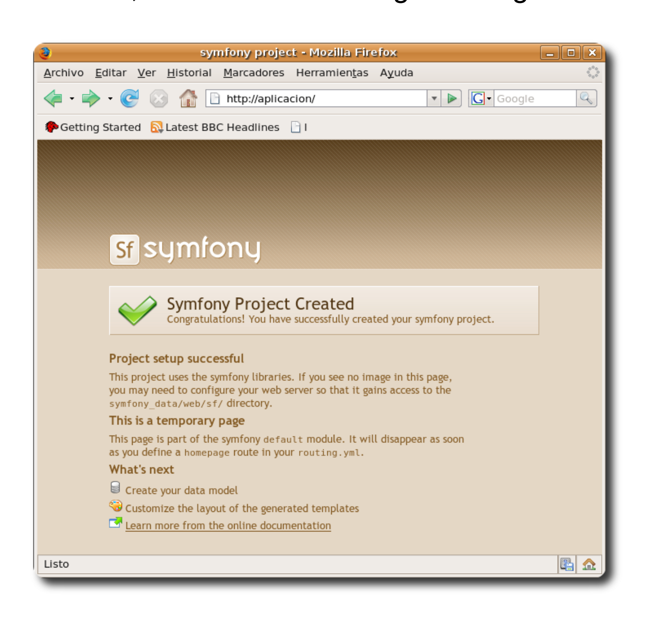


Gráfico 4a: Entorno de Producción

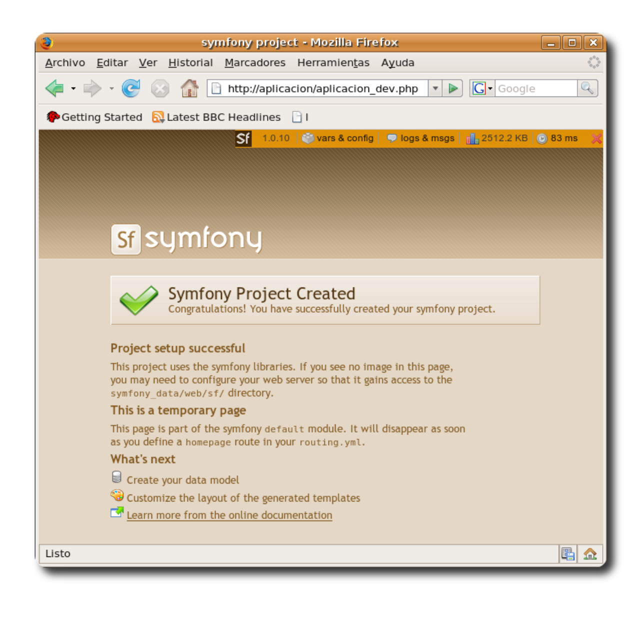


Gráfico 4b: Entorno de desarrollo

Modelado del dominio

Para describir el dominio de nuestra aplicación, symfony provee 2 alternativas:
Crear el esquema del dominio de forma manual.
Crear el esquema del dominio de forma automática desde una base de datos ya existente.
En este tutorial haremos uso de la primera alternativa; esta descripción debe ir incluida en el fichero config/schema.yml, así:

```
---
propel:
  evento:
     _attributes: { phpName: Evento }
       idMethod: native
     id: {type: INTEGER, required: true,
autoIncrement: true, primaryKey: true}
     nombre: {type: varchar, size: 50}
     ciudad: {type: varchar, size: 50}
     inicia: {type: date}
     finaliza: {type: date}
     sitioweb: {type: varchar, size: 50}
  expositor:
     _attributes: { phpName: Expositor }
       idMethod: native
     id: {type: INTEGER, required: true,
autoIncrement: true, primaryKey: true}
     paterno: {type: varchar, size: 25}
     materno: {type: varchar, size: 25}
     nombre: {type: varchar, size: 25}
     nacionalidad: {type: varchar, size: 25}
     email: {type: varchar, size: 50}
  conferencia:
     _attributes: { phpName: Conferencia }
     id: {type: INTEGER, required: true,
autoIncrement: true, primaryKey: true}
     evento_id:
     expositor_id:
     fecha: {type: date}
     titulo: {type: varchar, size: 50}
     resumen:{type: longvarchar}
```

Generación del modelo orientado a objetos

Symfony permite generar el modelo orientado a objetos en base al esquema (config/schema.yml), modelo que nos permitirá hacer uso de los conceptos y facilidades brindadas por el paradigma orientado a objeto. Esta generación se la realiza mediante la instrucción:

```
symfony propel-build-model
```

Generación del script SQL

Como en nuestra aplicación creamos de forma manual el esquema correspondiente al dominio de nuestra aplicación, debemos generar el script SQL para la creación de las tablas correspondientes en la base de datos utilizada en nuestra aplicación, en nuestro caso ejecutaremos este script en un servidor

mysql. Esta generación se la realiza mediante la instrucción:

```
symfony propel-build-sql
```

El archivo resultante esta almacenado en data/sql/lib.model.schema.sql

Acceso a la base de datos

Para que nuestra aplicación pueda acceder a la base de datos que contiene las tablas y datos, se deberá configurar el archivo config/databases.yml, de la siguiente forma:

```
all:
  propel:
    class:          sfPropelDatabase
    param:
      dsn:
mysql://usuario:clave@localhost/basededatos
```

Scaffolding

Scaffolding es un término introducido por Rails, para representar la generación de los artefactos que permiten un uso mas sencillo y rápido de las interfaces que permiten operaciones básicas, denominados de forma genérica CRUDs (Create, Read, Update y Delete) o ABMs (Altas Bajas y Modificaciones). El scaffolding aparte de crear las interfaces también crea los métodos que permiten acceder a los atributos de las distintas clases, con la posibilidad de ampliarlos y permitir mayor interacción y funcionalidad que la provista inicialmente.

Symfony permite 2 tipos de scaffolding:

- ✔ Orientadas a los usuarios del sistema.
- ✔ Orientadas a la administración de la aplicación

En este tutorial, haremos uso del scaffolding orientados a los usuarios del sistema, así como se muestra en la figura.

```
jesaavedra@zeus: ~/aplic_web/symfony/aplicacion
Archivo  Editar  Ver  Terminal  Solapas  Ayuda
jesaavedra@zeus:~/aplic_web/symfony/aplicacion$ symfony propel-generate-crud aplicacion evento Evento
>> dir+      /home/jesaavedra/aplic_web/symf...cacion/modules/evento/templates
>> file+     /home/jesaavedra/aplic_web/symf...vento/templates/editSuccess.php
>> file+     /home/jesaavedra/aplic_web/symf...vento/templates/showSuccess.php
>> file+     /home/jesaavedra/aplic_web/symf...vento/templates/listSuccess.php
>> dir+      /home/jesaavedra/aplic_web/symf...licacion/modules/evento/actions
>> file+     /home/jesaavedra/aplic_web/symf...vento/actions/actions.class.php
>> tokens    /home/jesaavedra/aplic_web/symf...vento/actions/actions.class.php
>> tokens    /home/jesaavedra/aplic_web/symf...vento/templates/editSuccess.php
>> tokens    /home/jesaavedra/aplic_web/symf...vento/templates/showSuccess.php
>> tokens    /home/jesaavedra/aplic_web/symf...vento/templates/listSuccess.php
>> tokens    /home/jesaavedra/aplic_web/symf...vento/actions/actions.class.php
>> file+     /home/jesaavedra/aplic_web/symf...plicacion/eventoActionsTest.php
>> tokens    /home/jesaavedra/aplic_web/symf...plicacion/eventoActionsTest.php
>> file-     /home/jesaavedra/aplic_web/symf...vento/actions/actions.class.php
>> dir-      /home/jesaavedra/aplic_web/symf...aa02d477467b/autoEvento/actions
>> file-     /home/jesaavedra/aplic_web/symf...vento/templates/listSuccess.php
>> file-     /home/jesaavedra/aplic_web/symf...vento/templates/showSuccess.php
>> file-     /home/jesaavedra/aplic_web/symf...vento/templates/editSuccess.php
>> dir-      /home/jesaavedra/aplic_web/symf...02d477467b/autoEvento/templates
>> dir-      /home/jesaavedra/aplic_web/symf...996082f9aa02d477467b/autoEvento
jesaavedra@zeus:~/aplic_web/symfony/aplicacion$
```

Gráfico 5: Scaffolding

De forma análoga se debe proceder a generar los scaffolds para todas clases de nuestro dominio, siempre que sean necesarias.

Ejecución y pruebas

Una vez corriendo nuestra aplicación podemos empezar a probar todas y cada una de las características provistas por los scaffolds generados:

- Formularios
- Ventanas de dialogo
- Relaciones
- Efectos JavaScript en ciertos campos
- ....

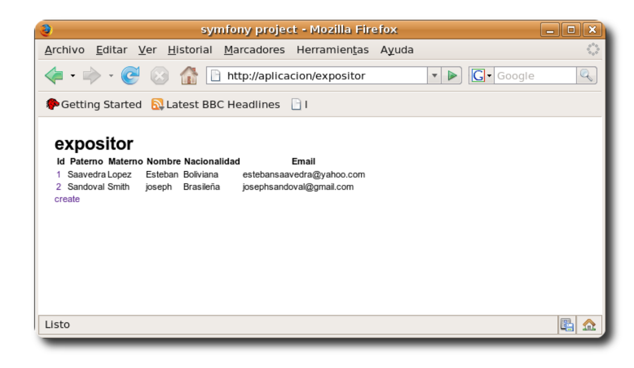


Gráfico 6: Listados

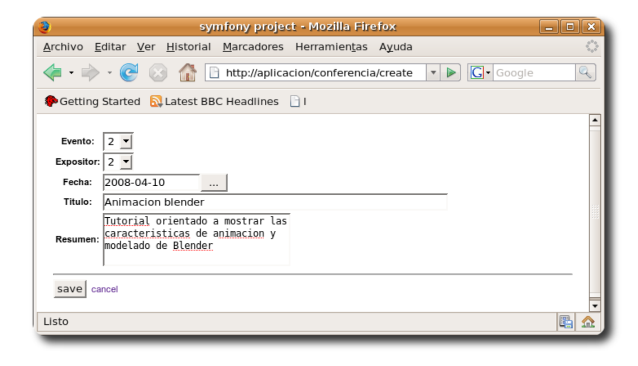


Gráfico 7: Formulario de entradas

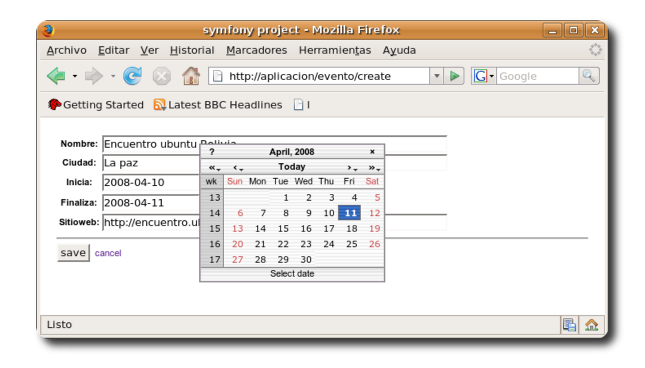


Gráfico 8: Efecto campo date

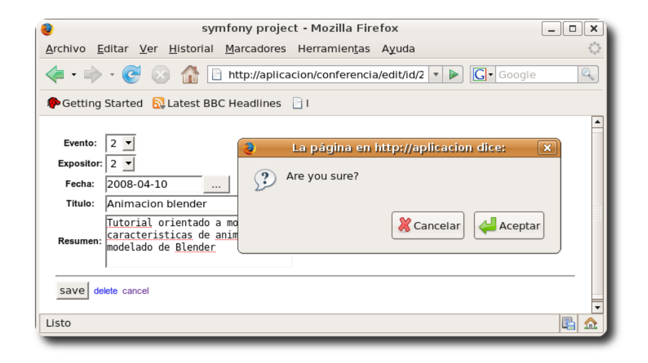
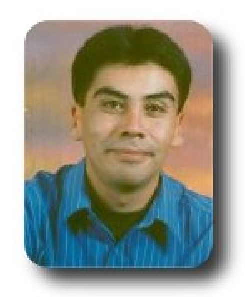


Gráfico 9: Ventanas de dialogo

Referencias

[1] http://www.symfony-project.org
[2] http://symfony.es
[3] The Definitive Guide to Symfony

Autor



Esteban Saavedra López
Líder de la Comunidad Atix (Oruro – Bolivia)
Activista de Software Libre en Bolivia
Email: jesaavedra@opentelematics.org
Website: http://jesaavedra.opentelematics.org


flisol
Festival Latinoamericano de Instalación
de Software Libre

2008

Presentación

El Festival Latinoamericano de Instalación de Software Libre (FLISoL) es el evento de difusión de Software Libre más grande en Latinoamérica. Se realiza desde el año 2005 y su principal objetivo es promover el uso del software libre, dando a conocer al público en general su filosofía, alcances, avances y desarrollo.

Para tal fin, las diversas comunidades locales de software libre (en cada país, en cada ciudad/localidad), organizan simultáneamente eventos en los que se instala, de manera gratuita y totalmente legal, software libre en las computadoras que llevan los asistentes. Además, en forma paralela, se ofrecen charlas, ponencias y talleres, sobre temáticas locales, nacionales y latinoamericanas en torno al Software Libre, en toda su gama de expresiones: artística, académica, empresarial y social.

¿Quién organiza el FLISOL2008?

La Comunidad de Software Libre en Latinoamérica, conformada por personas y grupos de diversa índole, con el apoyo de otras entidades, principalmente educativas, y algunos patrocinadores de los eventos en cada localidad.

ATIX ORURO Presente

La comunidad de Software Libre ATIX, también esta presente en este importante evento, cuyos detalles los puedes ver en: http://installfest.info/FLISOL2008/Bolivia/Oruro.

¿A quién está dirigido el evento?

El evento está dirigido a todo tipo de público: estudiantes, académicos, empresarios, trabajadores, funcionarios públicos, entusiastas y aun personas que no poseen mucho conocimiento informático.

¿Cuánto cuesta el evento?

La asistencia al evento es totalmente libre y gratuita.

¿Qué beneficios obtengo asistiendo al FLISOL2008?

¿Y qué pierdes?... Entre algunos beneficios, tendrás la oportunidad de instalar software libre en tu computadora, apreciar qué es una real y segura alternativa a otros modelos de desarrollo y distribución de software, informarte sobre la filosofía, cultura y organización alrededor del mismo, y conocer las experiencias, desarrollos e investigaciones de grupos y entidades en torno al Software Libre, tanto a nivel local, nacional e internacional.

Tal vez encuentres una alternativa/solución en software libre para ti, tu empresa, colegio, universidad e, incluso, para tu gobierno. Podrás aclarar tus dudas acerca de los temas relacionados y, quizás encuentres en el software libre, una alternativa de investigación, empleo y desarrollo tecnológico.

Detalles de FLISOL 2008 organizado por ATIX

Coordinadores

Esteban Saavedra L.
estebansaavedra@yahoo.com
http://jesaavedra.opentelematics.org

Williams Chorolque
williamsis@gmail.com

Colaboradores

Walter Cayoja
waltercayoja@gmail.com

Software disponible

- Ubuntu Bolivia
- CentOS 5.0, 5.1
- Opensuse
- Fedora 8
- FreBSD
- Knoppix 5.0
- Ubuntu 7.10
- Kubuntu 7.10
- Edubuntu 7.10
- Suse 10
- Debian etch
- FreeBSD 6.2
- FreeBSD 5.3
- PC-BSD
- Slackware 12
- Tuquito 2.0
- Ututo XS
- Mandriva 2007

Conferencias y Exposiciones

- Tuquito, dale una oportunidad (Oscar Choque Monzon)
- Desarrollo de aplicaciones en GNU/Linux con Qt (Williams Chorolque C.)
- Alternativas libres en un entorno Coorporativo (Esteban Saavedra L.)
- GNU/Linux para usuarios no técnicos (Victor Laura)
- Virtualbox: maquinas virtuales en GNU/LINUX (Daniel Aurelio Choque Chura)
- Aplicaciones Modernas del Código Abierto (Fernando Iquiza)

Nuestro afiche



El afiche nacional





Iniciamos con un vídeo Introductorio al **Software Libre**.



Williams Chorolque, mostrando las bondades de su Debian.



Daniel Choque haciendo una demostración de maquinas virtuales mediante VirtualBox.



Victor Laura comentando su experiencia, "*No se precisa ser informático para usar GNU/LINUX*", el estudia derecho.



Alegría y expectativa al probar e instalar algunas distribuciones.



Coordinadores y colaboradores de Flisol 2008. (*Williams, Esteban, Andy, Fernando, Juan C, Iver, Edson*), ...



Oscar Choque, demostrando las virtudes de Tuquito, una de las sensaciones del evento.



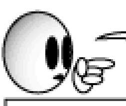
Fernando Iquiza mostrando el uso de GNU/Linux en dispositivos móviles.



Esteban Saavedra realizando la presentación de la revista **ATIX**.



Visita del director de Carrera de Ingeniería de Sistemas e Informática a **Flisol 2008**



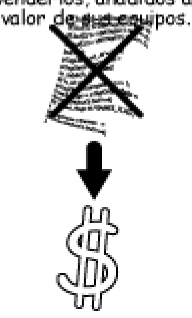
Historia de GNU/Linux.

Como ya dijimos, cada fabricante tenía su propio sistema operativo que servía sólo en sus máquinas. Entonces, Bell (o sea, AT&T) creo un sistema llamado UNIX. Y aquí se armó, porque UNIX era capaz de funcionar en muchas máquinas distintas.



Se hizo tremendamente popular no solamente por eso, sino también porque era estable y administraba bien los recursos de la computadora.

Eso hizo que las compañías se diesen cuenta del valor del software, y dejaron de dar el código fuente de sus programas y a venderlos, añadidos al valor de sus equipos.



Y ahora sí, volvemos al Sr. Stallman, que se enojó muchísimo cuando comprobó que el entorno en el que se desarrollaba el software era cada vez más cerrado.



El problema de no poder acceder al código fuente era que no se podía cambiar el funcionamiento de un programa para adaptarlo a lo que cada uno necesitase.



Como era bastante difícil que las compañías accediesen a dar el código fuente, decidió crear su propio sistema operativo y sus propios programas, poniendo a disposición siempre el código fuente. Este proyecto se llamó GNU.



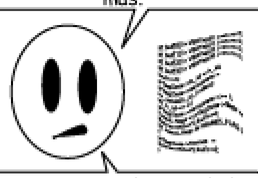
GNU es una especie de broma de moda en la época. Es un acrónimo recursivo que significa GNU's Not Unix, GNU no es Unix.

Pero lo más importante fue el concepto que Stallman presentó y que veremos más adelante: el software libre.



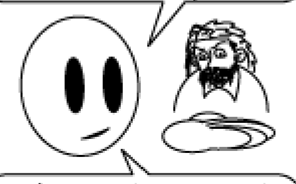
Este concepto interesó a muchos programadores y desarrolladores, que empezaron a involucrarse activamente en el proyecto de manera gratuita.

En 1984 se comenzó por desarrollar las herramientas necesarias para la realización del sistema operativo. El proyecto había calado hondo y la comunidad de GNU empezó a crecer más y más.



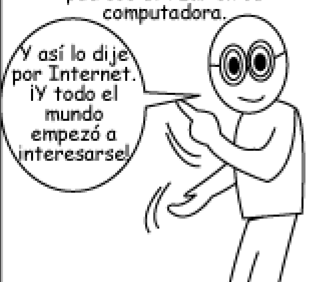
Poco a poco se dispuso de las herramientas necesarias para construir el núcleo de un sistema operativo. Se tenía pensado que este sistema se pareciese a UNIX debido a sus excelentes características.

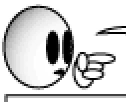
A estas alturas, Stallman había creado una organización para fomentar la creación de software libre: la FSF (Free Software Foundation)



Pero hacer un sistema operativo es mucho más complejo que crear los editores, compiladores, etc., necesarios para hacerlo. Sin embargo, resulta que Stallman no era el único en el mundo que había tenido la idea de hacer un sistema nuevo.

Aquí es donde entra un estudiante de la universidad de Helsinki llamado Linus Torvald, que decidió en 1991 crear para PC el núcleo de un sistema operativo como UNIX. Y, sobre todo, un núcleo que todo el que quisiera lo pudiese utilizar en su computadora.





Historia de GNU/Linux: distribuciones.

Entonces fue cuando se unieron los esfuerzos de Linus Torvalds, los de expertos en informática en todo el mundo y los del proyecto GNU y, utilizando las herramientas de estos últimos terminó naciendo... ¡Linux!

Este es Tux, la mascota-símbolo de Linux.



Es importante resaltar que el nombre completo es GNU/Linux por la manera en que se entrelazaron profundamente los dos proyectos.

El concepto detrás de GNU/Linux es el del software libre, también llamado copyleft, porque le da la vuelta al concepto de copyright.



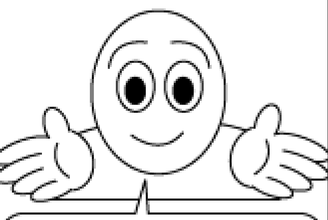
Acostumbrados a que cada compañía saque un sistema operativo y que vaya sacando diferentes versiones del mismo, puede ser un poco difícil comprender el concepto de las distribuciones de GNU/Linux.

Uno de los conceptos principales del software libre es que cada uno pueda modificar un programa como le de la gana sin tener que pedir permiso a nadie... y sin negarle ese permiso a nadie.



Por eso, los programas deben ir acompañados por su código fuente, o estar disponibles en un sitio dónde sea fácil de conseguir (Internet).

Otro concepto es que cada uno puede hacer lo que le de la gana con él: ofrecerlo sin costo, venderlo, alquilarlo... ¡y también sin dar cuenta a nadie!



O sea, yo escribo un programa y si quiero, puedo venderlo. Pero la persona que me lo compre, puede a su vez cambiarlo y volver a venderlo... o regalarlo.

Todas las partes de GNU/Linux se pueden descargar de Internet sin costo, pero para armarlas todas juntas es necesario saber mucho, muchísimo. Y tomarse su tiempo, claro, porque es un montón de trabajo. De ahí nacieron las distribuciones: recopilaciones de software libre interesante y útil (que puede ser cientos o miles de programas) que vienen con un instalador que permite dejar la computadora preparada para poder ser usada.

Vamos, lo de siempre: poner los CDs y ya seguir las instrucciones de la pantalla. ¡Y además vienen con unos programazos!

Existen compañías que se dedican a crear estas distribuciones y que las venden, pero no son muy caras, generalmente sólo cubren el material (los CDs, el empaque, los manuales, etc.). Hay otras que lo hacen sin ánimo de lucro, pero casi todas las distribuciones se pueden bajar de Internet gratuitamente.

Por cierto, puedes comprar cualquier distribución en tu puesto "pirata" más cercano... ¡porque copiar y vender una distribución no es delito!

Existen muchas distribuciones, tantas para cada gusto. Cada distribución tiene también diferentes versiones. La última siempre incluirá las últimas versiones no sólo del núcleo sino de los programas más importantes, como procesadores de texto, hojas de cálculo, etc. Aquí están algunas de las más conocidas:

Debian

Fedora Core

Slackware

Suse

A pesar de que GNU/Linux puede parecer que no es para todo el mundo, cada día es más fácil de usar. La mayoría de gente esta acostumbrada a Windows, pero las diferencias de manejo se van reduciendo cada vez más, a pesar de que Linux sigue siendo más complejo.



Aunque algunas distribuciones pueden dar problemas con algunos de los periféricos más modernos, las últimas versiones suelen remediarlos. Y, además de ser mucho más barato que Windows, GNU/Linux es mucho más estable y seguro.



Libera tus documentos



El Sajama



Ubicado al este de Bolivia, en el departamento de Oruro y muy cerca de la frontera con Chile, el Sajama está entre los 10 picos más altos de la Cordillera de los Andes. Con sus 6542 m.s.n.m. (21463 pies).



El Sajama fue montaña sagrada de la cultura preincaica de los Carangas, es la atracción principal del primer Parque Nacional de Bolivia que lleva su nombre (1939). En este parque de 100.223 hectáreas (247.653 acres) se encuentra un bosque del árbol que crece a mayor altura en el mundo, la queñua (de los 3600 a los 5200 m.s.n.m.).



Además, se encuentran tumbas de la cultura Chipaya y fauna endémica del lugar: aves, como flamencos, patos, gaviotas, gallinita de agua y avoceta andina y camélidos andinos, como llamas, alpacas, guanacos y vicuñas.

Esta ruta brinda un recorrido no sólo por el nevado Sajama, sino también por varias de las bellezas naturales del Parque Nacional que lleva su nombre.

Faro de Conchupata



Se puede disfrutar de una vista panorámica de la ciudad de Oruro desde el monumento nacional denominado Faro de Conchupata. Es un mirador extraordinario; se ubica justo en una elavación rocosa que servía de límite a las villas española e indígena. Es histórico por que allá se izó por primera vez la actual bandera boliviana.





Consultas

Contacto

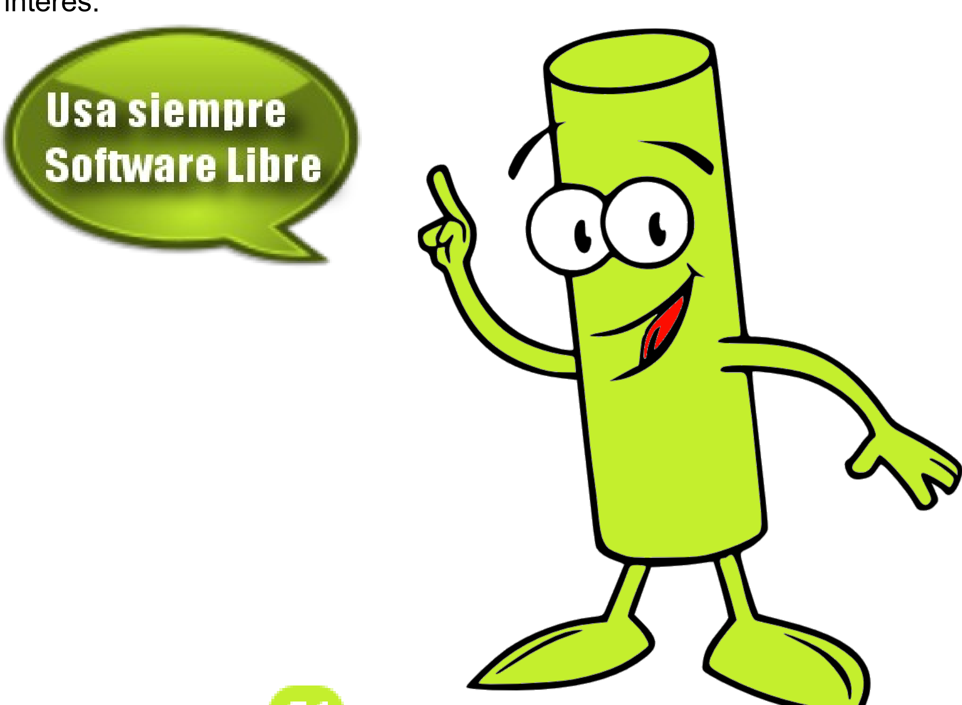
Para solicitar cualquier información, puedes contactar a:

- Esteban Saavedra López (jesaavedra@opentelematics.org)
- Williams Chorolque Choque (williamsis@gmail.com)

Publicación

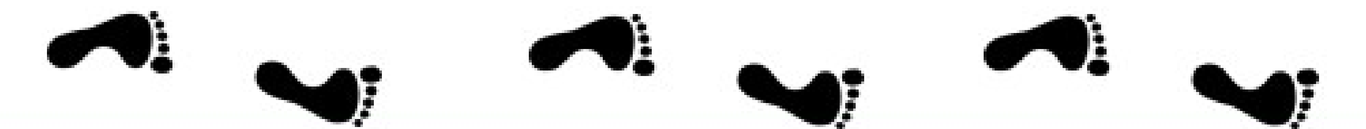
Te invitamos a ser parte de la **Revista ATIX**. La forma de participar puede ser enviándonos:

- Artículos referidos a áreas como:
  - Instalación y personalización de Aplicaciones
  - Scripting
  - Diseño gráfico
  - Programación y desarrollo de aplicaciones
  - Administración de servidores
  - Seguridad
  - y cualquier tema enmarcado dentro del uso de Software Libre
- Trucos y recetas.
- Noticias.
- Comics.
- Links de interés.





Marcamos Huella



http://atix.opentelematics.org